

10. csoport – 9. hét – 1. géptermi zh – 2014. április 8.

Feladat:

Poncius Pilátus ezen a héten is beszédet mond. A beszédet *szavanként* egy tömbben tároljuk. A beszéd hossza az „a”, „az” szavakat nem számolva is *legalább* 5 szó.

R betűt egy szava sem tartalmaz.

Kommunikációs szakembere javaslatára a selymesebb hangzás érdekében igyekezett a szavakban előforduló / betűket minél inkább a szavak első felébe sűríteni.

Állapítsuk meg, hogy *melyik* szó első felében (azaz a „közepe előtti betűig”) van a legtöbb / betű azon szavak közül, melyeknek a második felében (közepétől kezdve) már nincs / betű.

Pontozás:

- Sikeres beolvasás helyesen megadott adatok esetén fájlból vagy konzolról (tehát a bekérés módját Te döntöd el) – **10000 pont**
 - a tömb mérete előre *nem* ismert
 - fájlolvasásnál vagy a *fájl* végéig, vagy az *első* 'r' vagy 'R' karaktert tartalmazó szóig megy
 - konzolos beolvasásnál az *első* 'r' vagy 'R' karaktert tartalmazó szóig megy
 - ez a vizsgálat legyen külön *függvényben*, ami a mellékelt $l := \text{vanr}(s)$ nevű alprogram struktogramjának helyes átírása legyen.
- Beolvasás után annak ellenőrzése, valóban *legalább* 5 szó van-e a „a”-t és „az”-t nem számítva – **10000 pont**
 - konzolos bekérésnél hiba esetén a tömb *teljes* újra bekérése
 - fájlolvasásnál nem létező fájl vagy hiba esetén a fájl nevének újra bekérése és a tömb újbóli *teljes* beolvasása
 - az ellenőrzés kódjában jelenjen meg egy *számlálás* tétel (ehhez nem mellékeltem struktogramot)
- *Mindkét* beolvasási mód elkészítése, kettejük közül egy egyszerű menüvel lehessen választani – **10000 pont**
- A feladat *külső* és *két belső tételének* lekódolása – **25000 pont**
 - a 3 tételt mind külön *függvényként* írjátok meg
 - alternatív megoldást szabad írni, amennyiben az tételre visszavezetés eredményeképpen jön létre. Megfelelő paraméterezéssel és hívással ezt a feladatot meg lehet oldani mindössze két tétellel is! – *felt. maxker* és *számlálás*
- A főprogramban a feladatot megoldó függvény *meghívása* – **5000 pont**
 - az eredmény *értelmes* alakban történő kiírása, függően attól, hogy *van-e* megoldás

Lehet részpontokat kapni.

A jegy az elért pontok f függvénye. $f: [0..60000] \rightarrow \text{String}$, úgy hogy:

$\forall i \in [0..9999]: f(i) := \text{"dobjátok a padlóra!"}$

$\forall i \in [10000..19999]: f(i) := \text{"egyes"}$

$\forall i \in [40000..49999]: f(i) := \text{"négyes"}$

$\forall i \in [20000..29999]: f(i) := \text{"kettes"}$

$\forall i \in [50000..59999]: f(i) := \text{"ötös"}$

$\forall i \in [30000..39999]: f(i) := \text{"hármás"}$

$f(60000) := \text{"Centulio"}$

Jóóó munkát :F

A feltételes maximumkeresés specifikációja:

$$A = (t: String^n, van: \mathbb{L}, szo: String)$$

$$ef = \left(t = t' \wedge \sum_{\substack{i=1 \\ t[i] \neq "a" \wedge t[i] \neq "az"}}^n 1 \geq 5 \wedge \forall i \in [1..n]: ! vanr(t[i]) \right)$$

$$uf = \left(ef \wedge (van, ind) = \begin{matrix} n \\ MAX_{1,3} \\ i = 1 \\ nincsl(t[i]) \end{matrix} ldb(t[i]) \wedge (van \rightarrow szo = t[ind]) \right)$$

ahol:

$$vanr: String \rightarrow \mathbb{L}$$

$$vanr(s) = SEARCH_1 \begin{matrix} |s| \\ s_i = 'r' \vee s_i = 'R' \\ i = 1 \end{matrix}$$

$$nincsl: String \rightarrow \mathbb{L}$$

$$nincsl(s) = \begin{matrix} |s| \\ \forall SEARCH_1 \\ i = \lfloor |s|/2 \rfloor + 1 \end{matrix} s_i \neq 'l' \wedge s_i \neq 'L'$$

$$ldb: String \rightarrow \mathbb{N}$$

$$ldb(s) = \sum_{\substack{i=1 \\ s_i = 'l' \vee s_i = 'L'}}^{\lfloor |s|/2 \rfloor} 1$$

Algoritmus:

$van := \downarrow$				$i: \mathbb{N}$
$i = 1..n$				
$db, nincs := ldb(t[i]), nincsl(t[i])$				$db: \mathbb{N}, nincs: \mathbb{L} FHV$
$\neg nincs$	$nincs \wedge \neg van$	$nincs \wedge van$		
$SKIP$	$van, max, ind := \uparrow, db, i$	$db > max$		$max: \mathbb{N}$
		$max, ind := db, i$	$SKIP$	
van				
$szo := t[ind]$		$SKIP$		

Belső függvények:

$l := nincsl(s)$	
$l, i := \uparrow, \lfloor s /2 \rfloor + 1$	$i: \mathbb{Z}$
$l \wedge i \leq s $	
$l := s_i \neq 'l' \wedge s_i \neq 'L'$	
$i := i + 1$	

$db := ldb(s)$	
$db := 0$	$i: \mathbb{Z}$
$i = 1.. \lfloor s /2 \rfloor$	
$s_i = 'l' \vee s_i = 'L'$	
$db := db + 1$	
$SKIP$	

A beolvasás során meghívandó függvény:

$l := \text{vanr}(s)$	
$l, i := \downarrow, 1$	$i: \mathbb{Z}$
$\neg l \wedge i \leq s $	
$l := s_i = 'r' \vee s_i = 'R'$	
$i := i + 1$	

Tesztesetek:

A teszteseteket a zh-hoz mellékeltem, érdemes kipróbálni őket, egy jó visszajelzést nyújthatnak arról, hogy jól sikerült-e a kódolás. Ugyanakkor az értékelés a kód, és *nem* a futási eredmény alapján fog történni...

A teszteseteket *nem* lehet átszerkeszteni, a fájl (vagy a konzolra írt szöveg) formátuma kötött.

teszt1:

>>üres<<

→ nem elég hosszú a beszéd

teszt2:

az a az a az a az

→ nem elég hosszú a beszéd

teszt3:

az Loma a Lamszesz falao a lojalista

→ nem elég hosszú a beszéd (4 értelmes szó van benne, 5 kell)

teszt4:

a lomai szenatus es a romai nep neveben

→ nem elég hosszú a beszéd (mert csak 3 megfelelő szó van az első r betűs előtt!)

teszt5:

Jeluzsalem nepe! Loma a balatotok! Balati gesztuskent, ahogy ez mindig szokasos, elelesztunk egy labot a boltonbol! Melyik lab legyen a szelencses?

→ Jeluzsalem: nem számít, mert a „salem”-ben van l; nepe!: 0; Loma: 1; a: 0; balatotok!: 1; Balati: 1; gesztuskent,: 0; ahogy: 0; ez: 0; mindig: 0; szokasos,: 0; elelesztunk: 2 (első fele: elele, második fele: sztunk); egy: 0; labot: 1; a: 0; boltonbol!: nem számít, mert a „nbol!”-ban van l; Melyik: 1; lab: 1; legyen: 1; a: 0; szelencses?: 0. Ez alapján a megoldás: „elelesztunk”.

teszt6:

Dobjatok le a padlola kellek az onelzetes kis latolt!

→ Dobjatok: 0; le: 1; a: 0; padlola: „lola”-ban van l, ezért nem érdekes; kellek: „lek”-ben is van l; az: 0; onelzetes: „zetes”-ben nincs, ezért 1; kis: 0; latolt: „olt”-ban van. Megoldás: „le”