

Programozás Alapjai Projektfeladatok

Szoftverfejlesztés Tanszék

2013, ősz

Általános információk

A kötelező programot C nyelven kell megírni, és a *Bíró* webes felületén keresztül lehet benyújtani (<https://biro.inf.u-szeged.hu>). Egy C program kiterjesztése `c`. A *Bíró* a fájl nevében található első pont utáni részt tekinti kiterjesztésnek.

Kiértékelés

A kötelező programot a *Bíró* fogja kiértékelni. Feltöltés után a *Bíró* programot a `gcc` fordítóval `-static -O2` paraméterezéssel lefordítja, majd a programot 20 különböző tesztesetre futtatja. Minden helyes teszteset 1 pontot ér. A teszteset akkor helyes, ha a program futása nem tartott tovább 5 másodpercnél és a futása hiba nélkül (0 hibakóddal) fejeződött be és az adott inputhoz tartozó kimenet **minden egyes karaktere** megegyezik az előre eltárolt referencia megoldással. A programot 10 alkalommal lehet benyújtani, a megadott határidőig.

A *Bíró* által a `riport.txt`-ben visszaadott lehetséges hibakódok:

Futási hiba: 6	Memória- vagy időkorlát túllépés.
Futási hiba: 8	Lebegőpontos hiba, például nullával való osztás.
Futási hiba: 11	Memória-hozzáférési probléma, pl. tömb-túlinde克斯, null pointer használat.

Minden programra vonatkozó követelmények

A program bemenő adatait a `be.txt` nevű fájlból kell beolvasni, az eredményt pedig a `ki.txt` nevű fájlba kell írni akkor is, ha ez nincs külön megemlítve a feladat leírásában. A `be.txt` állomány csak olvasásra, a `ki.txt` állomány pedig csak írásra nyitható meg, más megnyitási mód esetén a *Bíró* nem engedélyezi a hozzáférést. Más fájl megnyitását a *Bíró* szintén nem engedélyezi.

A program bemenet/kimenet leírásokban a „sor” egy olyan karaktersorozatot jelöl, amelyben pontosan egy sorvége jel (`'\n'`) található, és az az utolsó karakter. Tehát minden sort sorvége jel zár! Elképzelhető olyan output, amelyben nincs sorvége jel, de akkor a feladat kiírásának egyértelműen jelznie kell, hogy a sorvége jel hiányzik!

A hibakód nélküli befejezést a `main` függvény végén végrehajtott `return 0;` utasítás biztosíthatja.

A feladatkiírással kapcsolatos megjegyzések

Igyekeztünk a megoldandó feladat leírását pontosná, a feladatot egyértelművé tenni. Ha ennek ellenére bármilyen pontatlanságot, kétértelműséget vagy hibát észlel a kiírásban, illetve bármilyen egyéb okból kérdése van a feladattal kapcsolatban, azt jelezze a gyakorlatvezetőjénél. A gyakorlatvezető a leadás előtti utolsó héten már nem válaszol a konkrétan a feladattal kapcsolatos kérdésekre.

[3305] Egyszerű tömörítő

Egy szöveget tömörítened kell, mégpedig a következő képpen: Keresd meg az öt leggyakoribb legalább 3 betűs szót a fájlban, és ezek helyére csak egy rövid kódot rakj! A kód két karakteres szó, a leggyakoribb szó kódja „!1” , a második leggyakoribb „!2” , ..., míg az ötödik leggyakoribb pedig „!5”. Azonos gyakoriság esetén a hamarabb előforduló szó számít gyakoribbnak. A többi szót hagyd meg, ahogy volt. A kódolt fájl elején fel kell sorolni a kódok jelentését.

A programnak dekódolnia is kell tudni. Így ha a bemenet a '!' karakterrel kezdődik, akkor be kell olvasni a kódokat és a hozzájuk tartozó szavakat, majd a szöveget visszaállítani az eredeti állapotába.

Bemenet

A **be.txt** fájl kétféle lehet. Az eredeti szöveg maximum 2000 db szót tartalmaz egy-egy szóközzel elválasztva egymástól, és az utolsó szó után nincs space, csak egy újsor karakter. A szavak maximum 22 karakteresek, és csak az angol ábécé kisbetűit tartalmazzák. Egyik szóban sem szerepel a '!' karakter. Mindig van legalább 5 fajta minimum hárombetűs szó a szövegben.

Figyelem! Lehet, hogy a bemeneti fájl formája nem az előbb leírtaknak megfelelő, hanem éppen a kimenetnél leírt formájú. Ilyenkor természetesen dekódolást a kell elvégezni. Dekódolás esetén a kódolásnál említett korlátok közül csak a 2000-es szószám (az 5 kód-szó páron túl) és 22 karakteres szóméret korlát érvényes.

Kimenet

A **ki.txt** fájl első öt sora tartalmazza a kódolási információkat. Mindegyikben az első szó a kód (pl.: „!1”), a második pedig az eredeti szó, amit helyettesít. Köztük egy szóköz van és a második szó után újsor karakter. Ezután pedig jön a kódolt szöveg, melyben nem kódolt szavak és kódok vegyesen követik egymást egy-egy szóközzel elválasztva. Az utolsó szó után csak egy újsor karakter van.

Figyelem! Lehet, hogy a kimeneti fájl formája nem az előbb leírtaknak megfelelő, hanem éppen a bemenetnél leírt formájú.

Példák

1. példa

Input

```
o xxa o xxb xxb o xxc o xxd xxb xxe xxe
```

Output

```
!1 xxb
!2 xxe
!3 xxa
!4 xxc
!5 xxd
o !3 o !1 !1 o !4 o !5 !1 !2 !2
```

2. példa

Input

```
!1 egy
!2 ketto
!3 harom
!4 negy
!5 ot
!1 a !2 b !3 !4 !5 c !1
```

Output

```
egy a ketto b három negy öt c egy
```

Segédanyag