Programozás Alapjai gyakorló ZH

1. feladatsor

Szoftverfejlesztés Tanszék

2023, Ősz

Általános információk

A programot C nyelven kell megírni, és a *Bíró* webes felületén keresztül lehet benyújtani. Egy C program kiterjesztése `c`. A *Bíró* a fájl nevében található első pont utáni részt tekinti kiterjesztésnek.

Kiértékelés

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

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-túlindexelés, 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 jeleznie kell, hogy a sorvége jel hiányzik!

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

1. feladat: Soronként cserélő kódolás és dekódolás (22 pont)

A kódoláshoz a szöveget sorfolytonosan adott számú oszlopba írjuk, majd az oszlopokat a kulcs által megadott sorrendben átrendezve az eredményt sorfolytonosan olvassuk ki. A dekódolás hasonló, csak a kulcs alapján vissza kell rendezni az oszlopokat. Például 7 oszloppal, 3 4 2 1 5 6 7 kulccsal és „eztaszovegetkelltitkositanod” szöveggel a táblázat csere előtt és után:

1	2	3	4	5	6	7	3	4	2	1	5	6	7
e	z	t	a	s	z	o	t	a	z	e	s	z	o
v	e	g	e	t	k	e	g	e	e	v	t	k	e
l	l	t	i	t	k	o	t	i	l	l	t	k	o
s	i	t	a	n	o	d	t	a	i	s	n	o	d

A titkosított szöveg: „tazeszogeevtketilltkotaisnod”.

Bemenet

A bemenetben található első szám mondja meg, hogy kódolásról (1) vagy dekódolásról (2) van szó. Utána következő szám az oszlopok száma. A második sorban található számok adják a kulcsot, vagyis hogy milyen sorrendben kell összekavarni az oszlopokat ahhoz, hogy a kódolt szöveget megkapjuk, illetve dekódolás esetén milyen sorrendben kell feltölteni a táblázat oszlopait ahhoz, hogy a sorfolytonos olvasással visszakapjuk az eredeti szöveget.

A harmadik sorban egy maximum 200 karakter hosszú szöveg található, ami nem tartalmaz szóközt és egyéb írásjeleket. Kódolás esetén ez a kódolandó szöveg, dekódolás esetén pedig a kódolt szöveg. A szöveg hossza maradék nélkül osztható az oszlopok számával, ezért a tömb minden sora teljes lesz. Az input fájl végén egy sortörés található. Az oszlopok száma legfeljebb 10 lehet.

Kimenet

A kimenet kódolás esetén a kulcs alapján kódolt szöveg, dekódolás esetén a dekódolt szöveg. A fájl végén egy sortörés található. Az input fájl tartalmának esetleges hibáit nem kell kezelni (pl. a kódolás/-dekódolás eldöntésére csak 1 vagy 2 szerepelhet).

Példák

1. példa

Input

```
1 7
3 4 2 1 5 6 7
eztaszovegetkelltitkositanod
```

Output

```
tazeszogeevtketilltkotaisnod
```

2. példa

Input

```
2 7
3 4 2 1 5 6 7
tazeszogeevtketilltkotaisnod
```

Output

```
eztaszovegetkelltitkositanod
```

Segédanyag

ASCII karakterkódok: `http://hu.wikipedia.org/wiki/ASCII`