

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-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 jeleznie kell, hogy a sorvége jel hiányzik!

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

1. feladat: Vigenére titkosítás és dekódolás (22 pont)

A feladat egy szöveg kulcs szerinti Vigenére titkosítása, illetve titkos szöveg dekódolása a kulcs alapján. Írjuk le a nyílt szöveget, majd írjuk fölé a kulcsszót ciklikusan ismételve. Alkalmazzuk minden nyílt szövegbeli betűre a felette lévő betű szerinti eltolást.

Például ha a kulcsszó: „EZAKULCSSZO”

eltolás	4	25	0	10	20	11	2	18	18	25	14	4	25	0	10	20	11	2
kulcs:	E	Z	A	K	U	L	C	S	S	Z	O	E	Z	A	K	U	L	C
nyílt sz.:	m	e	n	e	k	u	l	j	e	t	e	k	m	e	r	t	j	o
titkos sz.:	q	d	n	o	e	f	n	b	w	s	s	o	l	e	b	n	u	q

Vagyis az 'A' 0, a 'B' 1, ..., a 'Z' pedig 25-tel való eltolást jelent, és amennyiben az eltolás eredménye egy 'z' mögötti karakter lenne, úgy az ábécét újakezdjük az 'a' karakternél.

Bemenet

A bemenet első sorában az első karakter megmondja, hogy kódolásról (E) vagy dekódolásról (D) van szó, utána egy kettőspont, majd a legfeljebb 100 karakter hosszúságú, az angol ábécé nagybetűiből álló kulcs következik. A második sorban egy maximum 200 karakter hosszú szöveg található, ami csak az angol ábécé kisbetűit tartalmazza. Kódolás esetén ez a kódolandó szöveg, dekódolás esetén pedig a kódolt szöveg. Az input fájl végén egy sorvége jel található.

Kimenet

A kimenet kódolás esetén a kulcs alapján kódolt szöveg, dekódolás esetén a dekódolt szöveg. A fájl végén egy sorvége jel található. Az input fájl tartalmának esetleges hibáit nem kell kezelni (pl. a kódolás/dekódolás eldöntésére csak E vagy D karakter szerepelhet, az inputok szóközöket nem tartalmaznak, stb.).

Példák

1. példa

Input

```
E:EZAKULCSSZO
menekuljetekmertjonazellenseg
```

Output

```
qdn oefnbwssolebnuqfsyspkexmpi
```

2. példa

Input

```
D:EZAKULCSSZO
qdn oefnbwssolebnuqfsyspkexmpi
```

Output

```
menekuljetekmertjonazellenseg
```

Segédanyag

ASCII karakterkódok: <http://hu.wikipedia.org/wiki/ASCII>