

# Programozás Alapjai 10. ZH

## 14. 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: Oszloponként cserélő dekódolás (10 pont)

A dekódoláshoz az adott sorrendben töltjük fel az oszlopokat, majd a szöveget sorfolytonosan olvassuk ki a táblázatból, így visszanyerve az eredeti szöveget. Például 7 oszloppal, 3 4 2 1 5 6 7 kulccsal és „tgttaeiazeliavlsssttnzkkooeod” szöveggel a táblázat:

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

A dekódolt szöveg: „eztaszovegetkelltitkositanod”.

### Bemenet

A bemenetben található első szám  $K$  az oszlopok száma. A második sorban található számok adják a kulcsot, vagyis hogy milyen sorrendben kell feltölteni az oszlopokat ahhoz, hogy a sorfolytonos olvasással visszkapjuk az eredeti szöveget.

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

### Kimenet

A kimenet a dekódolt szöveg, vagy hibás input esetén egy hibaüzenet. Ha a kulcsban az  $[1, K]$  zárt intervallumon kívül eső oszlop számot találunk, akkor a kimenetbe kerüljön a „HIBA:OSZLOP: $N$ ” szöveg, ahol  $N$  a balról jobbra haladva talált első hibás oszlop érték. Ha a szöveg hossza nem osztható maradék nélkül az oszlopok számával, akkor a kimenetbe kerüljön a „HIBA:MARADEK” szöveg. A fájl végén egy sortörés található.

### Példák

#### 1. példa

Input

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

Output

```
eztaszovegetkelltitkositanod
```