

# Programozás Alapjai gyakorló ZH

## 1. feladatsor

Szoftverfejlesztés Tanszék

2022, 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-túlindekselés, 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: Egyábécés helyettesítő kódolás és dekódolás kulcs alapján (22 pont)

Adott egy 26 karakter hosszú kódoló vagy dekódoló kulcs, mely az angol ábécé betűinek permutációja. A kódoló kulcs csupa nagybetűt, a dekódoló kulcs csupa kisbetűt tartalmaz.

Az egyábécés helyettesítő kódolás során az angol ábécé kisbetűit a kódoló kulcs adott pozícióján lévő nagybetűvel helyettesítjük. Tehát az angol ábécé kisbetűit („abcdefghijklmnopqrstuvwxyz”) rendre helyettesítjük a kódoló kulcsban szereplő (pl. „DKVQFIBJWPESCXHTMYAUOLRGZN”) nagy betűkkel, vagyis, az 'a' betűket 'D'-kre, a 'b' betűket 'K'-ra, és így tovább lecseréljük. Ekkor megkapjuk a nagybetűkből álló kódolt szöveget.

A dekódolás során az angol ábécé nagybetűit a dekódoló kulcs adott pozícióján lévő kisbetűvel helyettesítjük. Tehát az angol ábécé nagybetűit („ABCDEFGHIJKLMNOPQRSTUVWXYZ”) rendre helyettesítjük a dekódoló kulcsban szereplő (pl. „sgmakexofhbwqzujdwlpctcinry”) kisbetűkkel, vagyis, az 'A' betűket 's'-re, a 'B' betűket 'g'-re, és így tovább lecseréljük. Ekkor megkapjuk a kisbetűkből álló dekódolt, eredeti szöveget.

A kódoló kulcs és a dekódoló kulcs egyértelműen megfeleltethető egymásnak, ezért elegendő csak az egyiket megadni, annak segítségével előállítható a másik kulcs is, így mindkét irányú művelet elvégezhető. Vagyis elképzelhető olyan eset, amikor a kulcs nagybetűs (kódoló kulcs), illetve a szöveg is nagybetűs (már kódolt). Ilyen esetben a már kódolt szövegből az eredeti szöveget kell visszanyerni, azaz dekódolni, viszont a kulcs azt jelzi, hogy a kódoláshoz mit használt a rendszer (a kódoló kulcs alapján a dekódoló kulcs is megkapható). Hasonlóan előfordulhat olyan eset, amikor a kulcs és a szöveg is kisbetűs (dekódoló kulcs és a már dekódolt üzenet alapján kell kódolni).

## Bemenet

A bemenet első sora tartalmazza a 26 karakterből álló kulcsot, ami dekódoló kulcs esetén az angol ábécé kisbetűiből, kódoló kulcs esetén a nagybetűiből áll (nem szerepelhetnek benne vegyesen kis- és nagybetűk). A második sorban található egy maximum 200 karakter hosszú szöveg, ami szintén az angol ábécé vagy csak nagy- vagy csak kisbetűit tartalmazhatja (és más karaktert nem tartalmaz). Kisbetűk esetén a feladat a kódolás, nagybetűk esetén a dekódolás. Tehát a műveletet a szöveg kis- vagy nagybetűssége határozza meg és NEM a kulcsé. Az input fájl végét sorvége jel zárja. Az, hogy a program kódoló vagy dekódoló kulcsot kap, és hogy a feladat kódolás vagy dekódolás, független egymástól!

## Kimenet

Kódolás esetén a kimenet a kódoló kulcs alapján kódolt szöveg csupa nagybetűkkel, dekódolás esetén a kimenet a dekódoló kulcs segítségével dekódolt kisbetűs szöveg. A fájl végén egy sorvége jel található. Az input fájl tartalmának esetleges hibáit nem kell kezelni (pl. vegyesen kis- és nagybetűs kulcsra vagy szóra nem kell felkészülni).

## Példák

### 1. példa

Input

```
DKVQFIBJWPESCXHTMYAUOLRGZN
ezamasodikkotelezoprogramomamikodolesdekodol
```

Output

```
FNDCDAHQWEEHUF SFNHTYHBYDCHCDCWEHQHSFAQFEHQHS
```

### 2. példa

Input

```
sgmakexofhbwqzujdwlpctcinry
FNDCDAHQWEEHUF SFNHTYHBYDCHCDCWEHQHSFAQFEHQHS
```

## Output

```
ezamasodikkkotelezoprogramomamikodolesdekodol
```

## Segédanyag

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