

Programozás Alapjai Nagy zh feladatok

Szoftverfejlesztés Tanszék

2014, ősz

Általános információk

A nagy zh feladatot 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 nagy zh feladatot a *Bíró* fogja kiértékelni. Feltöltés után a *Bíró* a 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 jeleznie 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 feladat beadása előtti utolsó héten már nem válaszol a konkrétan a feladattal kapcsolatos kérdésekre.

[4301] Számrendszerek közötti átváltás

A feladat adott számrendszerbeli számok átváltása másik számrendszerbe. Bemenetként adott az átváltandó szám számrendszerének alapja, melyik számrendszerbe kell átváltani, továbbá maga a szám. Figyeljünk oda, hogy csak akkor váltható át a megadott szám, ha az valóban a hozzá tartozó számrendszerben van ábrázolva, és ha a számrendszerek alapjai 2 és 36 közöttiek! A negatív számokkal nem kell foglalkozni. A szám értéke a $[0, 2^{31})$ tartományba esik.

Bemenet

A bemenetben három érték szerepel. Az első érték egy egész szám, mely az átváltandó szám számrendszerének alapja. A második érték szintén egy egész szám, mely az a számrendszer, amelybe át kell váltani a számot, és a harmadik érték maga az átváltandó nemnegatív egész szám az adott számrendszerben. A szám megadásánál a nem decimális karaktereket az angol ábécé 26 nagybetűje helyettesíti ('A' == 10, ..., 'Z' == 35). Mindhárom érték szóközzel van elválasztva egymástól, és a sor végén soremelés található. A számrendszerek alapjai $2 \leq alap \leq 36$ között lehetnek adottak. Ha ettől eltérő számrendszer van adva, akkor azt hibaként kell kezelni. Feltételezhető, hogy amennyiben a bemenetet nem hibaként kezeljük le, úgy az átváltandó szám a $[0, 2^{31})$ tartományba esik.

Kimenet

A kimenet az eredményt tartalmazza, vagyis az átváltott számot, esetleg a HIBA szót, ha valami miatt nem lehet átváltani a számot (pl. olyan számot adtunk meg, ami nem az első oszlopban megadott számrendszerben van ábrázolva, vagy ha valamelyik számrendszer nem 2 és 36 között van). A nem decimális karaktereket az angol ábécé 26 nagybetűje helyettesíti ('A' == 10, ..., 'Z' == 35). Az eredményt egy sorvége-karakter zárja!

Példák

1. példa

Input

10 16 255

Output

FF

2. példa

Input

3 10 Z

Output

HIBA

Segédanyag

Andrew S. Tannenbaum: *Számítógépes architektúrák*
Fried Ervin: *Oszthatóság és számrendszerek*

[4302] Tetszőleges és tízes számrendszerek közötti átváltás

A feladat adott számrendszerbeli számok átváltása tízes számrendszerbe és vissza. Bemenetként adott a számrendszer alapja, az átváltás iránya továbbá maga a szám. Figyeljünk oda, hogy egy szám csak akkor váltható át, ha valóban a hozzá tartozó számrendszerben van ábrázolva, és megadott számrendszer alapja 2 és 36 közötti! A negatív számokkal nem kell foglalkozni. A szám értéke a $[0, 2^{63})$ tartományba esik.

Bemenet

A bemenetben három érték szerepel. Az első két decimális alakban megadott érték között a „->” karakterek szerepelnek és az átváltás irányát jelzik. (Az egyik szám biztosan 10 lesz.) A nem tízes számrendszer alapja $2 \leq alap \leq 36$ között lehet. Ha ettől eltérő számrendszer van adva, akkor hibaként kell kezelni. A harmadik az átváltandó nemnegatív szám, amelyet az első („->” előtti) érték által meghatározott számrendszerből kell átváltani a másodikba. A harmadik érték szóközzel van elválasztva a másodiktól, és a sor végén sorvége-jel található. A szám megadásánál a nem decimális karaktereket az angol ábécé 26 kis- és nagybetűje helyettesíti ('a' == 'A' == 10, ..., 'z' == 'Z' == 35). Feltételezhető, hogy amennyiben a bemenetet nem hibaként kezeljük le, úgy az átváltandó szám a $[0, 2^{63})$ tartományba esik.

Kimenet

A kimenet az eredményt tartalmazza, vagyis az átváltott számot, esetleg a „HIBA” szót, ha valami miatt nem lehet átváltani a számot (pl. olyan számot adtunk meg, ami nem az átváltandó számrendszerben van ábrázolva, vagy ha valamelyik számrendszer nem 2 és 36 között van). A nem decimális karaktereket az angol ábécé 26 kisbetűje helyettesíti ('a' == 10, ..., 'z' == 35). Az eredményt egy sorvége-karakter zárja!

Példák

1. példa

Input

10->16 255

Output

ff

2. példa

Input

3->10 Z

Output

HIBA

Segédanyag

Andrew S. Tannenbaum: *Számítógépes architektúrák*
Fried Ervin: *Oszthatóság és számrendszerek*

[4303] Bináris és decimális számrendszerek közötti átváltás törttekkel együtt

A feladat bináris számok (kettedes törtek) átváltása decimális számokká (tizedes törtteké), és viszont. Tíz-es számrendszerből tizedes törtet 2-es számrendszerbe átváltva szintén törtet kapunk (kettedes), és fordítva is. Bemenetként adott a számrendszer alapja, továbbá a szám, amit át kell váltani. A negatív számokkal nem kell foglalkozni.

Bemenet

A bemenet egyetlen sorában két érték szerepel. Az első érték egy egész szám, mely az átváltandó szám számrendszerének az alapja (2 vagy 10). A második érték maga a szám, amit át kell váltani. A számérték lehet egész, ebben az esetben csak számjegyek szerepelnek benne. Ha törtről van szó, akkor mindenképpen van legalább egy számjegy a tizedespont előtt és után is. A két érték egymástól szóközzel van elválasztva, és a sor végén egy sortörés található.

A bináris számok számjegyeinek száma az egész és tört részben is külön-külön legfeljebb 60 lehet, a tízes számrendszerbeli számoknál pedig legfeljebb 18-18. Mindkét számrendszerbeli értéknél tizedespontot használunk. A bemenet esetleges hibáira (pl. más számrendszer, rossz számformátum) nem kell felkészülni.

Kimenet

A kimenet az eredményt tartalmazza a bemenethez hasonló formátumban. Vagyis elől a kimenet számrendszere, utána egy szóköz, majd az átváltott szám, és a végén egy sortörés. A tizedes törtet pontosan 15 tizedesjegy, a kettedes törtet 30 kettedesjegy pontossággal kell kiírni tizedespontot (kettedes pontot) használva. Ha a kimeneti számrendszerben a szám pontosan nem ábrázolható akkor lefelé kerekítést (csonkolást) kell alkalmazni. Egynél kisebb szám esetén a tizedespont előtt kötelező a 0.

Példák

1. példa

Input

```
10 0.4
```

Output

```
2 0.011001100110011001100110011001
```

2. példa

Input

```
2 101.0101010101010101010101010101010101010101
```

Output

```
10 5.333333333333332
```

Segédanyag

Andrew S. Tannenbaum: *Számítógépes architektúrák*
Fried Ervin: *Oszthatóság és számrendszerek*

[4304] Számolás valódi törtekkel, egyszerűsítéssel együtt

A feladat egy olyan program elkészítése, amely az inputból vett törtszámokon elvégzi az ugyancsak az inputban megadott műveletet. A művelet a szorzás, osztás, összeadás, kivonás valamelyike lehet. A művelet elvégzése után a törtet egyszerűsíteni kell.

Bemenet

A bemenetről (**be.txt**) tudjuk, hogy egyetlen sorból áll, benne két tört szám leírás, közöttük pedig egy műveleti jel (+, -, *, /). A tört szám leírás két (előjellel, vagy előjel nélkül megadott) egész számot tartalmaz egy kettősponttal (:) elválasztva, melyből az első a számláló, a második a nevező. A fentiekben és a sorvége karakteren kívül az input más karaktert nem tartalmaz. Feltételezhetjük, hogy bármely, az inputban kapott számláló és nevező értéke kisebb, mint 2^{31} és nagyobb, mint -2^{31} .

Amennyiben az inputban kapott egyik nevező értéke 0 (nulla), akkor az ahhoz tartozó műveletet nem kell elvégezni és az outputban egy törtszám helyett a „HIBA” szónak kell megjelennie (idézőjel és ékezetek nélkül, nagybetűvel, egy sorba írva). Más hibával nem kell számolni.

Kimenet

A művelet eredménye egy egyszerűsített törtszám, amelyet a **ki.txt** fájlba kell írni. A fájl egyetlen sort tartalmaz, amelybe kettősponttal elválasztva 2 egész számot írunk ki: a kettőspont előtt egy (negatív esetben előjeles) egész szám – az eredmény számlálója; a kettőspont után egy (előjeltelen) egész szám – az eredmény nevezője. A sort sorvége karakterrel zárjuk.

Az eredményt jelentő törtnek egyszerűsítve kell lennie! Ha az eredmény 0, akkor a számlálónak nullának kell lennie, a nevezőnek pedig 1-nek. Amennyiben az inputban bármely tört nevezője 0 (nulla) volt, akkor az ahhoz a törthöz tartozó műveletet nem kell elvégezni, hanem a művelet eredményeként az outputba a „HIBA” szónak (csupa nagybetűvel, ékezet nélkül, egy sorba írva) kell kerülnie.

Példák

1. példa

Input

5:-3+5:3

Output

0:1

2. példa

Input

5:-3*3:2

Output

-5:2

3. példa

Input

5:-3/3:0

Output

HIBA

Segédanyag

[4305] Oszloponként cserélő kódolás és dekódolás

A kódoláshoz a szöveget sorfolytonosan adott számú oszlopba írjuk, majd az oszlopokat a kulcs által megadott sorrendben olvassuk össze felülről lefelé. 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 „eztaszovegetkelltitkositanod” 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 titkosított szöveg: „tgttaeiazeliavlsssttnzkkooeod”.

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 összeolvasni az oszlopokat ahhoz, hogy a kódolt szöveget megkapjuk, illetve dekódolás esetén 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. 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

```
tgttaeiazeliavlsssttnzkkooeod
```

2. példa

Input

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

Output

```
eztaszovegetkelltitkositanod
```

Segédanyag

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

[4306] Soronként cserélő kódolás és dekódolás

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>

[4307] Vigenére titkosítás és dekódolás

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 szám megmondja, hogy kódolásról (1) vagy dekódolásról (2) van szó, az utána következő szám pedig a kulcs hosszát adja meg. A második sorban található a kulcs, ami az angol ábécé nagy betűiből állhat, legfeljebb 26 karakter hosszú lehet. A harmadik sorban egy maximum 200 karakter hosszú szöveg található, ami csak az angol ábécé kisbetűit tartalmazhatja. Kódolás esetén ez a kódolandó szöveg, dekódolás esetén pedig a kódolt szöveg. Az input fájl végén egy sortörés található.

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 11
EZAKULCSSZO
menekuljetekmertjonazellenseg
```

Output

```
qdn oefnbwssolebnuqfsyspkexmpi
```

2. példa

Input

```
2 11
EZAKULCSSZO
qdn oefnbwssolebnuqfsyspkexmpi
```

Output

```
menekuljetekmertjonazellenseg
```

Segédanyag

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

[4308] Vigenére titkosítás és dekódolás

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 újratekdjük 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

```
qdnofnbwssolebnuqfsyspkexmpi
```

2. példa

Input

```
D:EZAKULCSSZO
qdnofnbwssolebnuqfsyspkexmpi
```

Output

```
menekuljetekmertjonazellenseg
```

Segédanyag

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

[4309] Egyábécés helyettesítő kódolás és dekódolás kulcs alapján

Adott egy 26 karakter hosszú kulcs, mely az angol ábécé betűinek permutációja. Az egyábécés helyettesítő kódolás a kulcs alapján úgy történik, hogy az angol ábécé betűit a kulcs adott pozícióján lévő betűjével helyettesítjük. Tehát az angol ábécé betűit („abcdefghijklmnopqrstuvwxyz”) rendre helyettesítjük a kulcsban szereplő (pl. „dkvqfibjwpescxhtmyauolrgzn”) betűkkel, vagyis, az 'a' és 'A' betűket 'd' illetve 'D' betűkre, a 'b' és 'B' betűket 'k' illetve 'K' betűkre, és így tovább lecseréljük. Ekkor megkapjuk a kódolt szöveget. A kulcs segítségével a kódolt szöveg dekódolható ugyanazon kulcs alapján, ha az adott betűpárokat visszacseréljük.

Bemenet

A bemenet első sora tartalmazza a 26 karakterből álló kulcsot, ami az angol ábécé kis- vagy nagybetűiből áll. A második sorban található karakter mondja meg, hogy kódolásról ('K') vagy dekódolásról ('D') van szó. A harmadik sorban található egy maximum 255 karakter hosszú szöveg, ami csak az angol ábécé kis- és nagybetűit tartalmazza. Az input fájl végén egy sortörés található.

Kimenet

A kimenet kódolás esetén a kulcs alapján kódolt szöveg, dekódolás esetén a dekódolt szöveg. A két szövegben ugyanazon pozíción szerepelnek kis- és nagybetűk, vagyis a művelet a betű „nagyságán” nem változtat. A fájl végén egy sortörés található. Az input fájl tartalma a leírt formátumnak megfelelő lesz, annak esetleges hibáit nem kell kezelni (pl. a kódolás/dekódolás eldöntésére csak 'K' vagy 'D' szerepelhet).

Példák

1. példa

Input

```
DKVQFIBJWPESCXHTMYAUOLRGZN
K
ezamasodikkotelezoprogramomamiKODOLesDEKODOL
```

Output

```
fndcdahqweehufsfhntyhbydchdcwEHQHSfaQFEHQHS
```

2. példa

Input

```
DKVQFIBJWPESCXHTMYAUOLRGZN
D
fndcdahqweehufsfhntyhbydchdcwEHQHSfaQFEHQHS
```

Output

```
ezamasodikkotelezoprogramomamiKODOLesDEKODOL
```

Segédanyag

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

[4310] Egyábécés helyettesítő kódolás és dekódolás kulcs alapján

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. „sgmakexofhvbvquzjdwlpctinry”) 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.

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. 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

```
FNDCDAHQWEEHUFNFHTYHBYDCHCDCWEHQHSFAQFEHQHS
```

2. példa

Input

```
sgmakexofhvbvquzjdwlpctinry
FNDCDAHQWEEHUFNFHTYHBYDCHCDCWEHQHSFAQFEHQHS
```

Output

```
ezamasodikkotelezoprogramomamikodolesdekodol
```

Segédanyag

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