



## Titkos feladat

100 pont

Ilinca nagyon sokat olvasott az üzenetek titkosításáról, és most azt szeretné, ha a barátnőjével, Mirunával saját titkosítási rendszerrel kódolt üzeneteket váltanának.

Ilinca azt is tudja, hogy minden karaktert 8 biten ábrázolnak a számítógép memóriájában, ahol az adott karakter ASCII kódjának kettes számrendszerbeli alakja van eltárolva. A karakter titkosítására Ilinca egy négyzetes mátrixot használ, amelynek 8 sora (fentről lefele 0-tól 7-ig sorszámozva) és 8 oszlopa ( balról jobbra 0-tól 7-ig sorszámozva) van. A mátrix első sorába Ilinca a karakter ASCII kódjának kettes számrendszerbeli ábrázolását írja 8 biten, a 0-ik pozícióba a legszignifikánsabb számjegy kerül (a  $2^7$  -nek megfelelő). A mátrix következő 7 sorának mindegyikébe az előző sor elemeinek egy pozícióval balra történő körkörös permutációját írja. Ezután lexikografikus sorrendbe rendezi az így kapott mátrix sorait, majd úgy definiálja a karakter *titkos* alakját, mint egy olyan bitsorozatot, amelyet a mátrix utolsó oszlopának elemei alkotnak, fentről lefele haladva, ami után a mátrix azon sorának sorszáma következik, ahová a karakter ASCII kódjának kettes számrendszerbeli alakja került a lexikografikus rendezés során. Ha léteznek a mátrixban azonos tartalmú sorok, akkor a lexikografikus sorrendbe való rendezés után ezek megőrzik az eredeti relatív pozíciójukat, így az a sor, amelyben a karakter ASCII kódjának kettes számrendszerbeli alakja van, az első lesz ezek közül.

Ilinca úgy titkosít egy üzenetet, hogy egymás után írja az üzenetben szereplő karakterek *titkos* alakját. Miruna ismeri Ilinca titkosítási módszerét, ezért értelmezni is tudja a kapott titkosított üzeneteket.

### Követelmény

Írjatok programot, amely megoldja a következőket:

1. beolvas egy üzenetet, és kiírja az Ilinca módszerével titkosított üzenetet;
2. beolvas egy Ilinca módszerével titkosított üzenetet, és meghatározza a kítitkosított üzenetet.

### Bemeneti adatok

A `cript.in` bemeneti állomány első sorában egy  $c$  természetes szám van, amelynek értéke 1 vagy 2, aszerint, hogy melyik követelményt kell megoldani. Az állomány második sorában egy karakterlánc található.

### Kimeneti adatok

A `cript.out` kimeneti állományban egyetlen sor lesz, amelybe a bemeneti állományból vett szöveg titkosított alakját kell írni (ha  $c=1$ ), illetve a bemeneti állományból vett szöveg kítitkosított alakját (ha  $c=2$ ).

### Megkötések és pontosítások

- A nemtitkosított szöveg hossza nem nulla és nem nagyobb, mint 30000.
- A beolvasott szövegben szereplő karakterek kódja 32 és 127 közötti.
- Az  $s_1$  karakterlánc lexikografikus sorrendben az  $s_2$  karakterlánc előtt van, ha létezik egy olyan  $k$  pozíció, amelyre  $s_1[i]=s_2[i]$ , bármely  $i < k$  és  $s_1[k] < s_2[k]$  esetén.
- Az összpontszám 50%-ra, a tesztek az 1-es követelményre vonatkoznak.

### Példa

<code>cript.in</code>	<code>cript.out</code>	Magyarázatok
1 AB	100010004101000004	Az 'A' karakter ASCII kódja 65, a kettes számrendszerbeli alakja 8 biten 01000001. Baloldalt a körkörös permutáció után kapott mátrix látható, jobboldalt pedig az a mátrix, amelynek sorait lexikografikus sorrendbe rendeztük. <b>01000001</b> 00000101 10000010 00001010 00000101 00010100 00001010 00101000 00010100 <b>01000001</b> 00101000 01010000 01010000 10000010 10100000 10100000 Az 'A' ábrázolása a 4-es sorszámu sorban van. Az 'A' karakter <i>titkos</i> alakja: 100010004 Ugyanígy járunk el a 'B' esetében is
2 101110001111000002	VI	101110001-a 'V' karakter titkos alakja 111000002-az 'I' karakter titkos alakja

Maximális futási idő: 0.15 másodperc/teszt

Összmemória 2 MB, amelyből maximum 1 MB a verem mérete

A forráskód maximális mérete: 10 KB.