

## Problema 2 - Decod

100 puncte

Copilul lui Gigel, Costel, vrea să se joace de-a rețelistul. Astfel, acesta vrea să învețe cum sunt codificate, decodificate și verificate datele transmise printr-o rețea. El inventează o metodă ciudată prin care adaugă un bit de verificare în fața fiecărei perechi alăturate de octeți, de forma: primul octet cu al doilea octet, al treilea cu al patrulea, și așa mai departe. Acesta are valoarea 0 dacă lungimea celui mai lung subșir comun al celor doi octeți în reprezentare binară are lungimea pară, altfel are valoarea 1. Prin subșir al unui octet  $b=(b_1,b_2,\dots,b_8)$ , Costel înțelege un nou șir  $b'=(b_{i_1}, b_{i_2}, \dots, b_{i_K})$ , cu  $i_1 < i_2 < \dots < i_K$ . De exemplu, octetul  $b=(1, 0, 1, 1, 1, 0, 1, 1)$  conține ca subșir, șirurile  $(1, 1, 1, 1)$  sau  $(1, 0, 0, 1)$ , dar nu conține subșirul  $(0,0,1,1,1)$ .

### Cerințe

- 1) Codificați un mesaj dat din caractere ASCII în cod binar conform metodei inventate de Gigel.
- 2) Verificați dacă șirul binar dat a fost transmis corect.

### Date de intrare

Pe prima linie se va afla numărul natural  $q$  care poate avea valoarea 1 sau 2, în funcție de cerința problemei.

În cazul în care  $q$  are valoarea 1, pe următoarea linie se va afla mesajul de codificat. Altfel, dacă  $q$  are valoarea 2, pe următoarea linie se va afla mesajul binar de verificat.

### Date de ieșire

$q = 1$  : Mesajul codificat

$q = 2$  : Mesajul DA urmat pe a doua linie de mesajul decodificat dacă șirul binar este corect

Mesajul NU urmat pe a doua linie de numerele de ordine a perechilor de octeți care au bitul de verificare greșit. Indexarea perechilor de octeți de face de la 0.

### Restricții și precizări

- Textul dat pentru  $q = 1$  are lungimea maximă de 4000 de caractere
- Se garantează că textele de codificat au lungime pară
- Șirul dat pentru  $q = 2$  are lungimea maximă de 60000 de caractere
- Se garantează că șirul de biți dat are lungimea corectă, adică lungimea este multiplu de 17
- Pentru cerința 1 se acordă 30% din punctaj
- Pentru cerința 2 se acordă 30% pentru determinarea corectitudinii șirului binar și 40% pentru partea a doua a cerinței 2

Timp maxim de execuție/test: 0.1 secunde

Memorie totală disponibilă 2 MB, din care 1 MB pentru stivă

Dimensiunea maximă a sursei: 10KB

---

**Exemplu:**

decod.in	decod.out	Explicație
2 0010000100110010100010000001110011101110101011100100011 0010100100001	DA Be sure!	Toate codurile sunt corecte.

decod.in	decod.out	Explicație
2 0010010000110000100110100001100001	NU 1	A doua pereche de octeți are cel mai lung subșir comun 0110000 care are lungimea 7, ceea ce înseamnă că bitul de verificare ar fi trebuit să fie 1.

decod.in	decod.out	Explicație
1 Mama	0010011010110000100110110101100001	Ambele perechi de octeți au cel mai lung subșir comun de lungime pară, deci biții de verificare vor fi 0.

**Timp maxim de execuție/test: 0.1 secunde**

**Memorie totală disponibilă 2 MB, din care 1 MB pentru stivă**

**Dimensiunea maximă a sursei: 10KB**