

Problema 1 – paritate

100 puncte

În vederea asigurării unei transmiteri cât mai exacte a informațiilor pe rețea, transmiterea se efectuează caracter cu caracter, fiecare caracter fiind dat prin codul său ASCII, adică o grupă de 8 biți (octet). Pentru fiecare 8 biți transmiși se calculează un *bit de paritate* care are valoarea 0 (dacă codul ASCII al caracterului conține un număr par de cifre binare 1) sau 1 (în caz contrar). Deoarece în problema noastră se transmit numai caractere ASCII standard, cu codul ASCII din intervalul [32, 127], codul lor ASCII are bitul 7 (primul bit din stânga) egal cu 0. Pe această poziție va fi pus bitul de paritate, economisind astfel câte un bit pentru fiecare caracter transmis. De exemplu, dacă mesajul care trebuie transmis conține caracterele "Paritate", succesiunea de biți transmisă va fi:

01010000 11100001 01110010 01101001 01110100 11100001 01110100 01100101

În plus, pe lângă caracterele amintite, în mesaj mai poate să apară un caracter special, caracter care indică trecerea la începutul unui nou rând. Acest caracter are codul ASCII 10.

Cerință

Să se scrie un program care să verifice dacă un text a fost sau nu transmis corect.

Date de intrare

Fișierul de intrare `paritate.in` are pe prima linie o succesiune de caractere '0' și '1' care reprezintă mesajul transmis. Între caractere nu există spații. Linia se termină cu caracterul marcat de sfârșit de linie (`newline`).

Date de ieșire

Fișierul de ieșire `paritate.out` are pe prima linie mesajul **DA** dacă textul a fost transmis corect sau **NU** în caz contrar. În cazul în care mesajul de pe prima linie este **DA** liniile următoare vor conține textul transmis în clar. În cazul în care mesajul de pe prima linie este **NU** linia următoare va conține numerele de ordine ale caracterelor care nu au fost transmise corect, în ordine strict crescătoare, separate prin câte un spațiu.

Restricții și precizări

- Cei 8 biți ai codului ASCII a unui caracter se numerează de la 0 la 7, de la dreapta la stânga, cel mai din stânga bit fiind bitul 7 iar cel mai din dreapta bitul 0.
- Textul transmis are cel mult 60000 caractere.
- Numărul de caractere '0' și '1' din prima linie a fișierului de intrare este multiplu de 8.
- Codurile ASCII ale caracterelor din text aparțin mulțimii {10, 32-127}, codul 10 însemnând trecerea la începutul unui rând nou.
- Nici o linie din fișierul de ieșire nu va avea mai mult de 255 caractere.
- Caracterele din text sunt numerotate începând de la 0.
- mesajele DA/NU din prima linie a fișierului de ieșire se scriu cu majuscule.

Exemple

paritate.in	paritate.out	Explicație
0101000011100001011100100110100101110100111000010111010001100101	DA Paritate	Toate codurile sunt corecte

paritate.in	paritate.out	Explicație
1101000011100001111100100110100101110100111000010111010011100101	NU 0 2 7	Primul caracter a fost transmis ca succesiunea de biți 11010000 ceea ce înseamnă că fără bitul de paritate ar fi trebuit să existe un număr impar de cifre 1, ceea ce este fals. Deci caracterul nu a fost transmis corect. Același lucru se verifică și pentru caracterele cu numerele de ordine 2 și 7

paritate.in	paritate.out	Explicație
010000011111101001101001000010100110010100001010011010100110111101101001	DA A zi e j o i	Toate codurile sunt corecte. În text există două caractere cu cod ASCII 10

Timp maxim de execuție/test: 1 secundă